Package 'VertexWiseR'

November 10, 2026

Title Simplified Vertex-Wise Analyses of Whole-Brain and Subcortical Surface

Version 1.6.0

Date 2026-11-06

Maintainer Charly Billaud <charly.billaud@ntu.edu.sg>

Description Provides functions to run statistical analyses on surface-based neuroimaging data, computing measures including cortical thickness and surface area of the whole-brain and of the hippocampi. It can make use of 'FreeSurfer', 'fMRIprep', 'XCP-D', 'HCP' and 'CAT12' preprocessed datasets, 'HippUnfold' hippocampal outputs and 'SubCortexMesh' subcortical outputs for a given sample by restructuring the data values into a single file. The single file can then be used by the package for analyses independently from its base dataset and without need for its access.

License GPL-3

URL https://cogbrainhealthlab.github.io/VertexWiseR/

Suggests R.rsp

VignetteBuilder R.rsp

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

SystemRequirements Miniconda3 (v26.7.1), Python (<= v3.12), BrainStat (v0.4.2), vtk (v9.3.1), numpy (<= v1.26.4), nilearn (<= v0.12.0), netneurotools (<= v0.2.5)

Depends R (>= 4.0.0)

Imports doParallel, doSNOW, foreach, freesurferformats, fs, gifti, grDevices, grid, igraph, methods, mori, parallel, plotly, png, rappdirs, RColorBrewer, reticulate, stats, stringr, tools, utils

NeedsCompilation no

Author Junhong Yu [aut] (ORCID: <https://orcid.org/0000-0002-2563-9658>),
Charly Billaud [aut, cre] (ORCID:
<https://orcid.org/0009-0001-3466-9963>)

Repository CRAN

Date/Publication 2026-11-10 03:10:02 UTC

Contents

atlas_to_surf . . . 3
CAT12vextract . . . 4
decode_surf_data . . . 5
DTSERIESvextract . . . 6
edgelist-class . . . 7
edgelist_hip . . . 7
fs5_to_fs6 . . . 8
fs5_to_fslr . . . 8
fs6_to_fs5 . . . 9
fs6_to_fs5_map . . . 10
FSLRvextract . . . 10
fslr_to_fs5 . . . 12
fslr_to_fs5_map . . . 12
fs_stats . . . 13
getting_started . . . 13
HIPvextract . . . 14
hip_points_cells . . . 15
MNIsurface-class . . . 15
plot_overlay_surf . . . 16
plot_surf . . . 18
plot_surf3d . . . 20
RFT_vertex_analysis . . . 22
ROImap-class . . . 25
ROImap_fs5 . . . 25
ROImap_fs6 . . . 26
ROImap_fslr32k . . . 26
ROImap_hip . . . 27
SCMvextract . . . 27
smooth_surf . . . 29
SURFvextract . . . 30
surf_to_atlas . . . 31
surf_to_vol . . . 32
TFCE_threshold . . . 33
TFCE_vertex_analysis . . . 34
TFCE_vertex_analysis_mixed . . . 36
VWRfirstrun . . . 38

Index **40**

`atlas_to_surf` *Atlas to surface*

Description

Maps average parcellation surface values (e.g. produced with the surf_to_atlas() function) to the fsaverage5, fsaverage6 or fslr32k space

Usage

```
atlas_to_surf(parcel_data, template)
```

Arguments

`parcel_data`	A matrix or vector object containing average surface measures for each region of interest, see the surf_to_atlas() output format.
`template`	A string object stating the surface space on which to map the data ('fsaverage5', 'fsaverage6', 'fslr32k', 'CIT168' (hippocampal)).

Details

The function currently supports the Desikan-Killiany-70, Schaefer-100, Schaefer-200, Schaefer-400, Glasser-360, or Destrieux-148 atlases for cortical surfaces, and the 'bigbrain' 10-parcels atlas for hippocampal surfaces. ROI to vertex mapping data for 1 to 4 were obtained from the 'ENIGMA toolbox' ; and data for 5 from 'Nilearn' 's nilearn.datasets.fetch_atlas_surf_destrieux . atlas_to_surf() will automatically detect the atlas based on the number of columns.

Value

A matrix or vector object containing vertex-wise surface data mapped in fsaverage5, fsaverage6, fslr32k, or CIT168 space

See Also

`surf_to_atlas`

Examples

```
parcel_data = t(runif(100,min=0, max=100));
surf_data = atlas_to_surf(parcel_data, template='fsaverage5');
```

CAT12vextract *CAT12vextract*

Description

Extracts vertex-wise surface-based measures for each subject from a CAT12 preprocessed directory, resampled to a 32k mesh, and stores them all as a single .RDS file.

Usage

```
CAT12vextract(
  sdirpath = "./",
  filename,
  measure = "thickness",
  subj_ID = TRUE,
  silent = FALSE,
  VWR_check = TRUE
)
```

Arguments

sdirpath	A string object containing the path to the CAT12 subjects preprocessed directory. Default is the current working directory ("./").
filename	A string object containing the desired name of the output RDS file. Default is 'CAT12_measure.rds' in the R temporary directory (tempdir()).
measure	A string object containing the name of the measure of interest. Options are 'thickness', 'depth', 'fractaldimension', 'gyrification', and 'toroGI20mm'. Default is 'thickness.'
subj_ID	A logical object to determine whether to return a list object containing both subject ID and data matrix. Subject IDs are assumed to be the top directory names in the sdirpath.
silent	A logical object to determine whether messages will be silenced. Set to 'FALSE' by default
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Details

The function searches inside the CAT12 preprocessed for 32k meshes (.gii) with the user-selected measure, extracts the data from these files, and organizes the left and right hemisphere vertex data for each subject as rows in a N x 64984 data matrix within a .rds object. Python and reticulate are required as the NiBabel package is used to import .gii files outputted by CAT12.

Value

A .RDS file with a list containing 1. the list of subject IDs (first element) and 2. a surface data matrix object (second element), or only a data matrix object. The matrix has N subjects x V vertices dimensions and can be readily used by VertexWiseR statistical analysis functions. Each row corresponds to a subject (in the order they are listed in the folder) and contains the left to right hemispheres' vertex-wise values.

Examples

```
CAT12vextract(sdirpath="./",
filename='thickness.rds',
measure='thickness',
subj_ID = TRUE,
VWR_check=FALSE)
```

`decode_surf_data` *Decode surface data*

Description

Correlates the significant clusters of an earlier vertex-wise analysis with a database of task-based fMRI and voxel-based morphometric statistical maps and associate them with relevant key words. Decoding currently works with surfaces in fsaverage5 space only."

Usage

```
decode_surf_data(surf_data, contrast = "positive", VWR_check = TRUE)
```

Arguments

`surf_data`	A numeric vector or object containing the surface data, in fsaverage5 (1 x 20484 vertices) or fsLR32k (1 x 64984 vertices) space. It can only be one row of vertices (not a cohort surface data matrix).
`contrast`	A string object indicating whether to decode the positive or negative mask ('positive' or 'negative')
`VWR_check`	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Details

An internal image decoding is used, reproducing the 'NiMARE' python module's ROIAssociationDecoder in base R. The function also downloads the 'Neurosynth' database in the package's inst/extdata directory (converted to .RDS, ~7.2 MB) for the analysis.

Value

A data.frame object listing the keywords and their Pearson's R values

Examples

```
CTv = rbinom(20484, 1, 0.001)
decoding = decode_surf_data(CTv, 'positive', VWR_check=FALSE);
head(decoding)
```

DTSERIESvextract *DTSERIESvextract*

Description

Extracts vertex-wise surface-based CIFTI dense time-series data from an individual dtseries .nii file from HCP, fMRIprep or XCP-D preprocessed directories, and stores it as a single .RDS file.

Usage

```
DTSERIESvextract(dtseries, filename, silent = FALSE, VWR_check = TRUE)
```

Arguments

dtseries	A string object containing the full path to the dtseries files to extract from.
filename	A string object containing the desired name of the output RDS file. Default is 'fslr32k.rds' in the R temporary directory (tempdir()).
silent	A logical object to determine whether messages will be silenced. Set to 'FALSE' by default
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Details

The function extracts the data from the dtseries.nii file provided, and organizes the left and right hemisphere vertex data for each subject as rows in a N x 64984 data matrix within a .rds object.

Value

A .RDS file containing a surface data matrix object, with N time-point x V vertices dimensions and can be readily used by VertexWiseR statistical analysis functions. Each row corresponds to a time point in order and contains the left to right hemispheres' vertex-wise values.

Examples

```
#demo cifti dtseries from openneuro
#(ds005012, sub-18_ses-1_task-mid, run-01,
#reduced to 50 time points)
download.file(paste0("https://raw.githubusercontent.com/",
"CogBrainHealthLab/VertexWiseR/main/inst/demo_data/",
"demo_91k_bold.dtseries.nii"),
destfile=paste0(tempdir(),
```

```
"/demo_91k_bold.dtseries.nii"),
mode = "wb")

sub_dtseries=DTSERIESvextract(
dtseries=paste0(tempdir(),
            "/demo_91k_bold.dtseries.nii"),
silent=FALSE,
VWR_check=FALSE)

##visualizing e.g. the first 4 frames of the fMRI volume
#plot_surf(sub_dtseries[c(1,10,20,40),],
#            file="4frames.png")
```

`edgelist-class` *Edge list object*

Description

A class for adjacent vertex correspondance

Slots

`data` A N x 2 matrix object listing each vertex of the template and the vertices adjacent to it (making an edge together).

`name` The name of the template surface

`edgelist_hip` *List of edges for the hippocampal template*

Description

A Nx2 matrix object listing each vertex of the hippocampal template and the vertices adjacent to it (making an edge together).

Usage

```
edgelist_hip
```

Format

`edgelist_hip`**:**

Nx2 matrix object Matrix with two columns and N rows corresponding to the unique edges in the fsaverage5 surface

fs5_to_fs6 *fsaverage5 to fsaverage6*

Description

Remaps vertex-wise surface data in fsaverage5 space to fsaverage6 space using the nearest neighbor approach

Usage

```
fs5_to_fs6(surf_data)
```

Arguments

surf_data	A N x V matrix object containing the surface data (N row for each subject, V for each vertex), in fsaverage5 (20484 vertices) space. See also SURFvextract() output format.

Value

A matrix object containing vertex-wise surface data mapped in fsaverage6 space

See Also

fs6_to_fs5

Examples

```
CTv = runif(20484,min=0, max=100);
CTv_fs6 = fs5_to_fs6(CTv);
```

fs5_to_fslr *fsaverage5 to fslr*

Description

Remaps vertex-wise surface data in fsaverage5 space to fslr space using the nearest neighbor approach

Usage

```
fs5_to_fslr(surf_data)
```

Arguments

surf_data	A N x V matrix object containing the surface data (N row for each subject, V for each vertex), in fsaverage5 (20484 vertices) space. See also SURFvextract() output format.

Value

A matrix object containing vertex-wise surface data mapped in fslr space

See Also

fslr_to_fs5

Examples

```
CTv = runif(20484,min=0, max=100);
CTv_fslr = fs5_to_fslr(CTv);
```

`fs6_to_fs5` *fsaverage6 to fsaverage5*

Description

Remaps vertex-wise surface data in fsaverage6 space to fsaverage5 space using the nearest neighbor approach

Usage

```
fs6_to_fs5(surf_data)
```

Arguments

`surf_data`	A N x V matrix object containing the surface data (N row for each subject, V for each vertex), in fsaverage6 (81924 vertices) space. See also SURFvextract() output format.

Value

A matrix object containing vertex-wise surface data mapped in fsaverage5 space

See Also

fs5_to_fs6

Examples

```
surf_data = runif(81924,min=0, max=100);
fs5_data=fs6_to_fs5(surf_data)
```

fs6_to_fs5_map *fsaverage6 template object for nearest neighbor conversion in fs6_to_fs5()*

Description

fsaverage6 template object for nearest neighbor conversion in fs6_to_fs5()

Usage

```
fs6_to_fs5_map
```

Format

`fs6_to_fs5_map`**:**

An array of 81924 integers

vertices 81924 integers corresponding to each fsaverage6 vertex

FSLRvextract *FSLRvextract*

Description

Extracts vertex-wise surface-based measures for each subject from sMRIprep, fMRIprep, ASLprep, HCP processing or XCP-D preprocessed directories, and stores it as a single .RDS file.

Usage

```
FSLRvextract(
  sdirpath = "./",
  filename,
  dscalar,
  subj_ID = TRUE,
  silent = FALSE,
  VWR_check = TRUE
)
```

Arguments

`sdirpath`	A string object containing the path to the sMRIprep, fMRIprep, ASLprep, HCP preprocessed directory. Default is the current working directory ("./").
`filename`	A string object containing the desired name of the output RDS file. Default is 'fslr32k.rds' in the R temporary directory (tempdir()).

`dscalar` A string object containing the filename suffix of the dscalar file. These dscalar files are named differently depending on the preprocessing pipeline used. Examples of filename suffixes are shown below

- `.thickness_MSMAll.32k_fs_LR.dscalar.nii` (HCP MSMAll pipeline)
- `.sulc_MSMAll.32k_fs_LR.dscalar.nii` (HCP MSMAll pipeline)
- `.thickness.32k_fs_LR.dscalar.nii` (HCP legacy pipeline)
- `.sulc.32k_fs_LR.dscalar.nii` (HCP legacy pipeline)
- `_space-fsLR_den-91k_thickness.dscalar.nii` (sMRIprep/fMRIprep/ASLprep; using the `--cifti-output 91k` flag)
- `_space-fsLR_den-91k_curv.dscalar.nii` (sMRIprep/fMRIprep/ASLprep; using the `--cifti-output 91k` flag)
- `_space-fsLR_den-91k_sulc.dscalar.nii` (sMRIprep/fMRIprep/ASLprep; using the `--cifti-output 91k` flag)

`subj_ID` A logical object to determine whether to return a list object containing both subject ID and data matrix.

`silent` A logical object to determine whether messages will be silenced. Set to 'FALSE' by default

`VWR_check` A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Details

The function searches in the preprocessed directory by listing out files with certain suffixes, extracts the data from these files, and organizes the left and right hemisphere vertex data for each subject as rows in a N x 64984 data matrix within a .rds object.

Value

A .RDS file with a list containing 1. the list of subject IDs (first element) and 2. a surface data matrix object (second element), or only a data matrix object. The matrix has N subjects x V vertices dimensions and can be readily used by VertexWiseR statistical analysis functions. Each row corresponds to a subject (in the order they are listed in the folder) and contains the left to right hemispheres' vertex-wise values.

Examples

```
dat_fslr32k=FSLRvextract(sdirpath="./",
filename="dat_fslr32k.rds",
dscalar=".thickness_MSMAll.32k_fs_LR.dscalar.nii",
subj_ID = TRUE,
silent=FALSE,
VWR_check=FALSE)
```

fslr_to_fs5 *fslr to fsaverage5*

Description

Remaps vertex-wise surface data in fslr space to fsaverage5 space using the nearest neighbor approach

Usage

```
fslr_to_fs5(surf_data)
```

Arguments

surf_data	A N x V matrix object containing the surface data (N row for each subject, V for each vertex), in fslr (64984 vertices) space. See also SURFvextract() output format.

Value

A matrix object containing vertex-wise surface data mapped in fsaverage5 space

See Also

fs5_to_fslr

Examples

```
surf_data = runif(64984,min=0, max=100);
fs5_data=fslr_to_fs5(surf_data)
```

fslr_to_fs5_map *fslr template object for nearest neighbor conversion in fslr_to_fs5()*

Description

fslr template object for nearest neighbor conversion in fslr_to_fs5()

Usage

```
fslr_to_fs5_map
```

Format

fslr_to_fs5_map**:**
An array of 64984 integers

vertices 64984 integers corresponding to each fslr vertex

fs_stats *fs_stats()*

Description

Extracts descriptive statistics, for the whole-brain and subcortical region-of-interests (ROI), within a FreeSurfer subjects directory. It reads them from the aseg.stats file, as generated by the default FreeSurfer preprocessing pipeline.

Usage

```
fs_stats(sdirpath = "./", sublist, ROImeasure = "Volume_mm3")
```

Arguments

sdirpath	A string object indicating the path to the 'FreeSurfer' subjects directory. Default is the current working directory ("./").
sublist	A string object indicating the path to the subject list generated by SURFvextract as 'sublist.txt' (optional). This allows users to retrieve stats only from a selected list of subjects. The subject list is a list with 1 subject ID per line.
ROImeasure	A string object indicating what summary measure to extract for the subocrtical ROIs. Choices include: 'NVoxels', 'Volume_mm3', 'StructName', 'normMean', 'normStdDev', 'normMin', 'normMax', and 'normRange'. Default is 'Volume_mm3'.

Value

A data.frame object with N columns per aseg.stats measures and N row per subjects.

Examples

```
fs_stats(sdirpath="freesurfer_subjdir")
```

getting_started *Getting started with tutorials*

Description

Guides the user through the various tutorials and assist with downloading the demo data

Usage

```
getting_started(demo_data = TRUE)
```

Arguments

demo_data	A boolean object specifying whether to prompt user about downloading the demo data (default is TRUE)

Examples

```
getting_started()
```

HIPvextract *HIPvextract*

Description

Extracts hippocampal vertex-wise surface-based measures for each subject in the 'HippUnfold' subjects directory, and stores it as a single .RDS file.

Usage

```
HIPvextract(
  sdirpath = "./",
  filename,
  measure = "thickness",
  subj_ID = TRUE,
  silent = FALSE
)
```

Arguments

sdirpath	A string object containing the path to the 'HippUnfold' subjects directory. Default is the current working directory ("./").
filename	A string object containing the desired name of the output RDS file. Default is 'hip_measure.rds' in the R temporary directory (tempdir()).
measure	A string object containing the name of the measure of interest. Options are 'thickness','curvature','gyrification' and 'surfarea' (For more information see the 'HippUnfold' documentation). Default is thickness.
subj_ID	A logical object stating whether to return a list object containing both subject ID and data matrix.
silent	A logical object to determine whether messages will be silenced. Set to 'FALSE' by default

Details

The function searches for the hippocampal surface data by listing out files with certain suffixes, extract the data from these files, and organize the left and right hippocampal vertex data for each subject as rows in a N x 14524 data matrix within a .rds object.

Value

A .RDS file with a list containing 1. the list of subject IDs (first element) and 2. a surface data matrix object (second element), or only a data matrix object. The matrix has N subjects x V vertices dimensions and can be readily used by VertexWiseR statistical analysis functions. Each row corresponds to a subject (in the order they are listed in the folder) and contains the left to right hemispheres' hippocampal vertex-wise values.

Examples

```
HIPvextract(sdirpath = "./", filename = paste0(tempdir(),"/hip_data.RDS"), measure = "thickness")
```

---

hip_points_cells *points and cells data required to build the hippocampus surface template*

Description

points and cells data required to build the hippocampus surface template

Usage

```
hip_points_cells
```

Format

`hip_points_cells`**:**

A list object with two data frame objects:

vertices data frame with 7262 rows (vertices), 3 columns (MNI coordinates X, Y, Z)

vertices data frame with 14266 rows (vertices), 3 columns (vertices of all unique triangles

vertices data frame with 7262 rows (vertices), 3 columns (MNI coordinates X, Y, Z for unfolded hippocampal surface)

MNIsurface-class *MNI surface map object*

Description

A class for surface coordinates in MNI space

Slots

`data` A matrix object with N vertices columns x 3 rows corresponding to each vertex's X,Y,Z coordinates in MNI space.

`name` The name of the template surface

`plot_overlay_surf` *Surface overlay plotter*

Description

Plots surface data in a grid with one row in a .png file as in plot_surf(), with parameters to overlay two maps on top of eachother.

Usage

```
plot_overlay_surf(
  model_output = NULL,
  surf_data_1 = NULL,
  surf_data_2 = NULL,
  cmap_1,
  cmap_2,
  limits_1,
  limits_2,
  alpha_1 = 1,
  alpha_2 = 1,
  colorbar_1 = TRUE,
  colorbar_2 = TRUE,
  show_nan = TRUE,
  filename,
  title = "",
  surface = "inflated",
  overlay_boundaries = FALSE,
  size,
  zoom,
  transparent_bg = FALSE,
  smooth_mesh = 0,
  show.plot.window = TRUE,
  VWR_check = TRUE
)
```

Arguments

`model_output`	A list object outputted by RFT_vertex_analysis() or TFCE_threshold(). The 'tstat_map' will automatically be treated as background map, and the 'thresholded_tstat_map' as overlay map. See surf_data_1 or surf_data_2 to assign any map manually.
`surf_data_1`	A numeric vector (length of V), where V is the number of vertices. It can be one row from the output from SURFvextract(), CAT12vextract(), FSLRvextract(), HIPvextract(), SCMvextract(), as well as masks or vertex-wise results outputted by analyses functions. This is the background surface.
`surf_data_2`	Same as surf_data_1. This is the overlay surface.

`cmap_1` A string object specifying the name of an existing colormap or a vector of hexadecimal color codes to be used as a custom colormap. The names of existing colormaps are listed in the 'Matplotlib' plotting library.

Default cmap is set to `"Reds"` for positive values, `"Blues_r"` for negative values and `"RdBu"` when both positive and negative values exist. This is meant for the background surface.

`cmap_2` Same as cmap_1. This is meant for the overlay surface. Default is the same as cmaps_1, adapted to limits of surf_data_2.

`limits_1` A combined pair of numeric vector composed of the lower and upper color scale limits of surf_data_1. When left unspecified, the symmetrical limits c(-max(abs(surf_data_1),max(abs(surf_data_1))) will be used. If set to NULL, the limits will correspond to the min and max values of surf_data_1. This is meant for the background surface.

`limits_2` Same as limits_1, or also the string object "same". Default is symmetrical limits relative to surf_data_2, while 'same' will apply the exact same limits as for limits_1. This is meant for the overlay surface.

`alpha_1` A numeric object between 0 and 1 to determine the opacity of the non-empty vertices. Note that this is not a true opacity setting, it will blend the colour into that of the NaN vertices (white if show_nan is FALSE). This is meant for the background surface.

`alpha_2` Same as alpha_2. This is meant for the overlay surface. Default is the same as alpha_1.

`colorbar_1` A logical object stating whether to include the color bar for the background layer in the plot or not (default is TRUE).

`colorbar_2` A logical object stating whether to include the color bar for the overlay layer in the plot or not (default is TRUE).

`show_nan` A logical object to determine if the NaN vertices are to be plotted (Default is TRUE). This is meant for the background surface. The overlay surface will always omit NaN vertices to make the background visible.

`filename` A string object containing the desired name of the output .png. Default is 'combined_plots.png' in the R temporary directory (tempdir()). Only filenames with a .png extension are allowed.

`title` A string object for setting the title in the plot. Default is none. For titles that too long to be fully displayed within the plot, we recommend splitting them into multiple lines by inserting "\n".

`surface` A string object containing the name of the type of cortical surface background rendered. Possible options include "white", "smoothwm","pial" and "inflated" (default).

`overlay_boundaries` A logical object stating whether to plot black contour of the overlay layer.

`size` A combined pair of numeric vector indicating the image dimensions (width and height in pixels). Default is c(1700,400) for cortical surfaces and c(860,200) for Hippunfold hippocampal surfaces. Note that the size will depend on the inclusion of color bar(s), which will expand the width to ~5% per color bar.

zoom	A numeric value for adjusting the level of zoom on the figures. Default is 1.25 for cortical surfaces and 1.2 for Hippunfold hippocampal surfaces.
transparent_bg	A logical object to determine if the background of the image is set to transparent (Default is FALSE).
smooth_mesh	A numeric object stating the number of iterations of smoothing to make the surface appear smoother. Default is 0.
show.plot.window	A logical object to determine if the generated plot is to be shown within RStudio's plot window
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Value

Outputs the plot as a .png image

Examples

```
#simulate t-map
background=rep(NA,20484);
ROImap_fs5 <- get('ROImap_fs5')
ROImap <- list(ROImap_fs5@data,ROImap_fs5@atlases)
neg_parts <- c(1:100)
idx_neg <- which(ROImap[[1]][,1] %in% neg_parts)
background[idx_neg]=runif(length(idx_neg),-1,0)
pos_parts <- c(18:46)
idx_pos <- which(ROImap[[1]][,1] %in% pos_parts)
background[idx_pos]=runif(length(idx_pos),0,1)
#simulate clusters (randomly picked aparc atlas parcels)
idx_sig=which(ROImap[[1]][,1]==29| ROImap[[1]][,1]==19|
            ROImap[[1]][,1]==45)
overlay = rep(NA,20484);
overlay[idx_sig]=1

plot_overlay_surf(surf_data_1=background,
                surf_data_2=overlay,
                cmap_1='RdBu_r', cmap_2='Reds',
                alpha_1=0.4, alpha_2=1,
                filename=paste0(tempdir(),"/simulated_plot.png"),
                title="Significant effects",
                overlay_boundaries=TRUE,
                VWR_check=FALSE)
```

---

plot_surf *Surface plotter*

Description

Plots surface data in a grid with one or multiple rows in a .png file

Usage

```
plot_surf(
  surf_data,
  filename,
  title = "",
  surface = "inflated",
  cmap,
  limits,
  colorbar = TRUE,
  size,
  zoom,
  transparent_bg = FALSE,
  show_nan = T,
  alpha = 1,
  smooth_mesh = 0,
  show.plot.window = TRUE,
  VWR_check = TRUE
)
```

Arguments

surf_data	A numeric vector (length of V) or a matrix (N rows x V columns), where N is the number of subplots, and V is the number of vertices. It can be the output from SURFvextract(), CAT12vextract(), FSLRvextract(), HIPvextract(), SCMvextract(), as well as masks or vertex-wise results outputted by analyses functions. Alternatively, atlas ROI values as supported by atlas_to_surf() may be given.
filename	A string object containing the desired name of the output .png. Default is 'plot.png' in the R temporary directory (tempdir()).Only filenames with a .png extension are allowed.
title	A string object for setting the title in the plot. Default is none. For titles that too long to be fully displayed within the plot, we recommend splitting them into multiple lines by inserting "\n".
surface	A string object containing the name of the type of cortical surface background rendered (only applicable on fsaverage5 and fsaverage6). Possible options include "white", "smoothwm","pial" and "inflated" (default).
cmap	A string object specifying the name of an existing colormap or a vector of hexadecimal color codes to be used as a custom colormap. The names of existing colormaps are listed in the 'Matplotlib' plotting library. Default cmap is set to "Reds" for positive values, "Blues_r" for negative values and "RdBu" when both positive and negative values exist.
limits	A combined pair of numeric vector composed of the lower and upper color scale limits of the plot. If the limits are specified, the same limits will be applied to all subplots. When left unspecified, the same symmetrical limits c(-max(abs(surf_dat),max(abs(surf_dat))) will be used for all subplots. If set to NULL, each subplot will have its own limits corresponding to their min and max values

colorbar	A logical object stating whether to include a color bar in the plot or not (default is TRUE).
size	A combined pair of numeric vector indicating the image dimensions (width and height in pixels). Default is c(1920,400) for whole-brain surface and c(400,200) for HippUnfold hippocampal surface.
zoom	A numeric value for adjusting the level of zoom on the figures. Default is 1.25 for whole-brain surface and 1.20 for HippUnfold hippocampal surface.
transparent_bg	A logical object to determine if the background of the image is set to transparent (Default is FALSE).
show_nan	A logical object to determine if the NaN vertices are to be plotted (Default is TRUE).
alpha	A numeric object between 0 and 1 to determine the opacity of the non-empty vertices. Note that this is not a true opacity setting, it will blend the colour into that of the NaN vertices (white if show_nan is FALSE)
smooth_mesh	A numeric object stating the number of iterations of smoothing to make the surface appear smoother. Not applicable for HippUnfold hippocampal data. Default is 0.
show.plot.window	A logical object to determine if the generated plot is to be shown within RStudio's plot window
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Value

Outputs the plot as a .png image

Examples

```
results = runif(20484,min=0, max=1);
plot_surf(surf_data = results,
filename=paste0(tempdir(),"/output.png"),
title = 'Cortical thickness',
surface = 'inflated', cmap = 'Blues',
VWR_check=FALSE)
```

---

plot_surf3d *3D Surface plotter*

Description

Plots 3D cortical surfaces

Usage

```
plot_surf3d(
  surf_data,
  surf_color = "grey",
  cmap,
  limits,
  atlas = 1,
  hemi = "b",
  medial_gap = 0,
  orientation_labels = TRUE,
  plot_grid = TRUE,
  transparent_bg = FALSE,
  smooth_mesh = 0,
  VWR_check = TRUE
)
```

Arguments

surf_data	A numeric vector (length of V), where V is the number of vertices. It can be the output from SURFvextract(), CAT12vextract(), FSLRvextract(), HIPvextract(), SCMvextract().
surf_color	color of the cortical surface for NA values. Set to 'grey' by default. A RGBA string can also be given, with A as the opacity (0 to 1), e.g. for transparent grey: surf_color="rgba(100,100,100,0.5)".
cmap	A string vector containing 2 to 4 color names/codes specifying the colors to be used for the color scale; or a single string object with the name of a color map listed in RColorBrewer::display.brewer.all(). If none are specified, appropriate colors will be automatically selected according to range(surf_data)
limits	A combined pair of numeric vector composed of the lower and upper color scale limits of the plot. When left unspecified, the symmetrical limits c(-max(abs(surf_dat),max(abs(surf_ will be used.
atlas	atlas used for identifying region labels. 1=Desikan, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400. Set to 1 by default. This argument is ignored for HippUnfold hippocampal surfaces. For applicable SubCortexMesh surfaces, 1=default base ROI, 2=anatomical atlas.
hemi	A string specifying the hemisphere to plot. Possible values are l (left), r (right) or b (both).
medial_gap	A numeric value specifying the amount of gap (in MNI coordinate units) to separate the left and right hemispheres. Set to 0 (no gap between hemispheres) by default. In order to view the medial surfaces clearly, it is recommended that this value is set to 20. This argument is ignored if hemi!='b'
orientation_labels	A boolean object specifying if orientation labels are to be displayed. Set to TRUE by default
plot_grid	A boolean object specifying whether to plot the orientation grid or not (default is TRUE).

transparent_bg	A boolean object specifying whether to get a transparent background upon saving the image (default is FALSE, white background).
smooth_mesh	A numeric object stating the number of iterations of (Laplacian) smoothing to make the surface appear smoother. Not applicable for HippUnfold hippocampal data. Default is 0.
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Value

a plot_ly object

Examples

```
surf_data = runif(20484);
plot_surf3d(surf_data = surf_data, VWR_check=FALSE)
```

RFT_vertex_analysis *Vertex-wise analysis with random field theory cluster correction*

Description

Fits a linear or linear mixed model with the cortical or hippocampal surface data as the predicted outcome, and returns cluster-thresholded (Random field theory) t-stat map selected contrast.

Usage

```
RFT_vertex_analysis(
  model,
  contrast,
  random,
  formula,
  formula_dataset,
  inverse = FALSE,
  surf_data,
  p = 0.05,
  atlas = 1,
  smooth_FWHM,
  VWR_check = TRUE
)
```

Arguments

model	An N X P data.frame object containing N rows for each subject and P columns for each predictor included in the model. This data.frame should not include the random effects variable.

`contrast` A N x 1 numeric vector or object containing the values of the predictor of interest. Its length should equal the number of subjects in model (and can be a single column from model). The cluster-thresholded t-stat maps will be estimated only for this predictor.

`random` A N x 1 numeric vector or object containing the values of the random variable (optional). Its length should be equal to the number of subjects in model (it should NOT be inside the model data.frame).

`formula` An optional string or formula object describing the predictors to be fitted against the surface data, replacing the model, contrast, or random arguments. If this argument is used, the formula_dataset argument must also be provided.

- The dependent variable (DV) is not needed, and the formula will start with ~. The DV will be the surface data value by default, but it can be swapped with contrast as IV via the "inverse" argument.
- The first independent variable in the formula will always be interpreted as the contrast of interest for which to estimate cluster-thresholded t-stat maps.
- Only one random regressor can be given and must be indicated as '(1|variable_name)'.

`formula_dataset` An optional data.frame object containing the independent variables to be used with the formula (the IV names in the formula must match their column names in the dataset).

`inverse` A boolean object stating whether to set the surface data as predictor of the contrast variable, instead of as dependent variable (default is FALSE). Other covariates in the model remain independent variables. This makes modelling slower.

`surf_data` A N x V matrix object containing the surface data (N row for each subject, V for each vertex), in fsaverage5 (20484 vertices), fsaverage6 (81924 vertices), fslr32k (64984 vertices) or hippocampal (14524 vertices) space. See also Hipvextract(), SURFvextract() or FSLRvextract output formats. Alternatively, a string object containing the path to the surface object (.rds file) outputted by extraction functions may be given.

`p` A numeric object specifying the p-value to threshold the results (Default is 0.05)

`atlas` An integer corresponding to the atlas of interest. 1=Desikan, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400. Set to 1 by default. This argument is ignored for hippocampal surfaces. For applicable SubCortexMesh surfaces, 1=default base ROI, 2=anatomical atlas.

`smooth_FWHM` A numeric vector object specifying the desired smoothing width in mm. It should not be specified if the surf_data has been smoothed previously with smooth_surf(), because this results in surf_data being smoothed twice.

`VWR_check` A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Details

The function imports and adapts the 'BrainStat' Python library.

By default, false discovery rate correction is used together with the Random field theory (RFT) cluster correction. To look at data without any form of cluster correction, users can simply refer to the outputted 'tstat_map'.

Output definitions:

- `nverts`: number of vertices in the cluster
- `P`: p-value of the cluster
- `X, Y and Z`: MNI coordinates of the vertex with the highest t-statistic in the cluster.
- `tstat`: t-statistic of the vertex with the highest t-statistic in the cluster
- `region`: the region this highest -statistic vertex is located in, as determined/labelled by the selected atlas

Value

A list object containing the cluster level results, unthresholded t-stat map, thresholded t-stat map, positive, negative and bidirectional cluster maps, and a FDR-corrected p-value map.

See Also

`TFCE_vertex_analysis`, `TFCE_vertex_analysis_mixed`

Examples

```
demodata = readRDS(system.file('demo_data/SPRENG_behdata_site1.rds',
package = 'VertexWiseR'))[1:100,]
CTv = readRDS(file = url(paste0("https://github.com",
"/CogBrainHealthLab/VertexWiseR/blob/main/inst/demo_data/",
"SPRENG_CTv_site1.rds?raw=TRUE")))[1:100,]

vertexwise_model=RFT_vertex_analysis(model=demodata[,c("sex","age")],
contrast=demodata[,"age"], surf_data = CTv, atlas=1,p = 0.05,
VWR_check=FALSE)

#Description of the output:
#vertexwise_model$cluster_level_results

#Formula alternative:
#formula= as.formula("~ age + sex")
#vertexwise_model=RFT_vertex_analysis(formula=formula,
#formula_dataset=demodata, surf_data = CTv, atlas=1, p = 0.05,
#VWR_check=FALSE)
```

ROImap-class *Region-of-Interest mapping object*

Description

A class for surface vertices mapping on atlas labels

Slots

`data` A matrix object with N vertices from a template and each parcellation number the vertices correspond to in 6 atlases (6 columns).

`atlases` Each of the 6 available atlases and their corresponding labels (1=aparc, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400).

`name` The name of the template surface

ROImap_fs5 *Atlas parcellations of fsaverage5*

Description

A list containing two data frames, 1) listing fsaverage5 vertices and each parcellation number they correspond to, and 2) listing each available atlas and their corresponding labels (1=aparc, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400).

Usage

```
ROImap_fs5
```

Format

`ROImap_fs5`**:**

A list object with two data frame objects:

vertices data frame with 20484 rows (vertices), 6 columns (atlases)

atlases data frame with 400 rows (labels, not all are filled depending on atlas), 6 columns (atlases)

ROImap_fs6 *Atlas parcellations of fsaverage6*

Description

A list containing two data frames, 1) listing fsaverage6 vertices and each atlas parcellation number they correspond to, and 2) listing each available atlas and their corresponding labels (1=aparc, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400).

Usage

```
ROImap_fs6
```

Format

`ROImap_fs6`**:**

A list object with two data frame objects:

vertices data frame with 81924 rows (vertices), 6 columns (atlases)

atlases data frame with 400 rows (labels, not all are filled depending on atlas), 6 columns (atlases)

ROImap_fslr32k *Atlas parcellations of FS_LR32k*

Description

A list containing two data frames, 1) listing FS_LR32k vertices and each atlas parcellation number they correspond to, and 2) listing each available atlas and their corresponding labels (1=aparc, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400).

Usage

```
ROImap_fslr32k
```

Format

`ROImap_fslr32k`**:**

A list object with two data frame objects:

vertices data frame with 64984 rows (vertices), 6 columns (atlases)

atlases data frame with 400 rows (labels, not all are filled depending on atlas), 6 columns (atlases)

ROImap_hip *Atlas parcellations of the hippocampus (CITI168)*

Description

A list containing 1) a matrix listing CITI168 vertices and each parcellation number they correspond to, and 2) a data frame listing the hippocampal atlas labels.

Usage

```
ROImap_hip
```

Format

`ROImap_hip`**:**

A list object with two data frame objects:

vertices array of 14524 numeric vectors (vertices)

atlases data frame with 10 rows listing names of left and right hippocampal subfields

SCMvextract *SCMvextract*

Description

Extracts surface-based measures from the python toolbox "SubCortexMesh" in each subject. These can be based on FreeSurfer ASeg or FSL FIRST subcortical segmentations, as well as CIFTI subcortical grayordinates. SCMvextract() reads through the metrics .vtk files outputted by SubCortexMesh (surface_metrics/ or surface_metrics_cifti/ directory), extracts the vertex-wise values, and stores them as a .rds file, merging left and right hemispheres when applicable, for each (selected) subcortical region.

Usage

```
SCMvextract(
  sdirpath = "./",
  outputdir,
  template,
  measure = "thickness",
  roilabel,
  subj_ID = TRUE,
  silent = FALSE,
  VWR_check = TRUE
)
```

Arguments

`sdirpath`	A string object containing the path to the 'SubCortexMesh' surface metrics directory, containing subjects subdirectories, each with surface-based metrics for each region (.vtk files). Default is the current working directory ("./").
`outputdir`	A string object containing the path of the directory where all ROI RDS files will be stored. Default is 'subcortices' in the R temporary directory (tempdir()).
`template`	A string object containing the name of the template segmentation which was used in SubCortexMesh ('fsaverage' or 'fslfirst').
`measure`	A string object containing the name of the measure of interest. Options include thickness, curvature, surfarea and functional metrics generated by SCM's cifti_metrics() function. Default is thickness.
`roilabel`	A string object or vector of string objects containing the name(s) of the subcortical region(s) to extract (both hemispheres automatically included when applicable). E.g. "thalamus" or c("thalamus", "caudate"). Default is all regions in sdirpath. Note that if roilabel is undefined and the SubCortexMesh output contains merged surface objects such as "allaseg" or "allfslfirst", they will also be extracted along with the individual regions. `roilabel='allaseg'` will only extract the merged object.
`subj_ID`	A logical object to determine whether to return a list object containing both subject ID and data matrix.
`silent`	A logical object to determine whether messages will be silenced. Set to 'FALSE' by default
`VWR_check`	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Details

The function looks for standard subject folder names (starting "sub-") and surface file names as defined by SubCortexMesh (starting "left-","right-", or "brain-stem"). SCMvextract() requires reticulate and python to read the .vtk files.

Number of vertices in the (bilateral) matrix for each region-of-interest:

- fsaverage / fslfirst: ROI
- 2044 / 2026: accumbens area
- 3430 / 3592: amygdala
- 6940 / 7570: caudate nuclei
- 39214 / 31466: cerebellum
- 8132 / 8244: hippocampus
- 3200 / 3548: pallidum
- 8394 / 7908: putamen
- 7768 / 8542: thalamus
- 7144 / NA: ventral diencephalon
- 9452 / 9516: brain stem
- 95718 / 82412: all ROIs merged

Value

A directory containing - for each pair of bilateral subcortical regions separately - .RDS files, each with a list containing 1. the list of subject IDs (first element) and 2. a surface data matrix object (second element), or only a surface data matrix object. Each matrix has N subjects x V vertices dimensions and can be used readily by VertexWiseR statistical analysis functions. Each row corresponds to a subject (in the order they are listed in the folder) and contains the left to right hemispheres' vertex-wise values (except for the Brain Stem).

Examples

```
SCMvextract(sdirpath = "subcortexmesh_output_metrics",
            outputdir=paste0(tempdir(), "\\subcortices"),
            template='fsaverage',
            measure="surfarea",
            VWR_check=FALSE)
```

---

smooth_surf *Smooth surface*

Description

Smooths surface data at defined full width at half maximum (FWHM) as per the corresponding template of surface data

Usage

```
smooth_surf(surf_data, FWHM, VWR_check = TRUE)
```

Arguments

surf_data	A N x V matrix object containing the surface data (N row for each subject, V for each vertex), in fsaverage5 (20484 vertices), fsaverage6 (81924 vertices), fslr32k (64984 vertices) or hippocampal (14524 vertices) space. See also SCMvextract() for subcortical regions. Alternatively, a string object containing the path to the surface object (.rds file) outputted by extraction functions may be given.
FWHM	A numeric vector object containing the desired smoothing width in mm
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Value

A matrix object with smoothed vertex-wise values

Examples

```
surf_data = readRDS(file = url(paste0("https://github.com",
"/CogBrainHealthLab/VertexWiseR/blob/main/inst/demo_data/",
"FINK_Tv_ses13.rds?raw=TRUE")))[1:3,]

surf_data_smoothed=smooth_surf(surf_data, 10, VWR_check=FALSE);
```

SURFvextract *SURFvextract*

Description

Extracts whole-brain vertex-wise surface-based measures for each subject in a 'FreeSurfer' output subjects directory, resamples the data to a common surface template, and stores it as a .rds file. This function requires the 'FreeSurfer' environment to be preset in the unix environment and a 'FreeSurfer' license key.

Usage

```
SURFvextract(
  sdirpath = "./",
  filename,
  template = "fsaverage5",
  measure = "thickness",
  subj_ID = TRUE,
  fshomepath
)
```

Arguments

sdirpath	A string object containing the path to the 'FreeSurfer' preprocessed subjects directory. This directory must be the output directory from a FreeSurfer preprocessing recon-all pipeline. Default is the current working directory ("./").
filename	A string object containing the desired name of the output RDS file. Default is 'brain_measure.rds' in the R temporary directory (tempdir()).
template	A string object containing the name of surface template (available: 'fsaverage5', 'fsaverage6'). Default is fsaverage5.
measure	A string object containing the name of the measure of interest. Options include thickness, curv, sulc, area, and volume (for freesurfer 7.4.1 or later). Default is thickness. Note that non-standard surface measures stored in FreeSurfer's 'surf/' directories also work, provided the full name and extension following lh./rh. is given (e.g., "w-g.pct.mgh" for grey-white matter contrast).
subj_ID	A logical object stating whether to include subject IDs (folder names in the subjects directory) as a first column to the output matrix. Default is TRUE.
fshomepath	An optional string object containing the path to the FreeSurfer installation directory. This makes sure R accesses FreeSurfer if the system environment variables are not inherited — as would be the case if you are running the function from RStudio.

Details

Note that RStudio does not inherit the shell environment variables if it is open from a terminal. In that case, the fshomepath argument needs to be provided. The function runs system shell commands that will produce in the set subjects directory: 1) a sorted list of subjects "sublist.txt"; 2) a link file to the selected surface fsaverage template. 3) left and right hemisphere .mgh maps outputted by 'FreeSurfer' 's mris_preproc.

Value

A .RDS file with a list containing 1. the list of subject IDs (first element) and 2. a surface data matrix object (second element), or only a data matrix object. The matrix has N subjects x V vertices dimensions and can be used readily by VertexWiseR statistical analysis functions. Each row corresponds to a subject (in the order they are listed in the folder) and contains the left to right hemispheres' vertex-wise values.

Examples

```
SURFvextract(sdirpath = "freesurfer_subjdir",
filename=paste0(tempdir(), "/CTv.rds"), template="fsaverage5",
measure="curv")
```

---

surf_to_atlas *Surface to atlas*

Description

Returns the mean or sum of vertex-wise surface data for each ROI of a selected atlas

Usage

```
surf_to_atlas(surf_data, atlas, mode = "mean")
```

Arguments

surf_data	A N x V matrix object containing the surface data (N row for each subject, V for each vertex), in fsaverage5 (20484 vertices), fsaverage6 (81924 vertices), fslr32k (64984 vertices) or hippocampal (14524 vertices) space. See also Hipvextract(), SURFvextract() or FSLRvextract() output formats.
atlas	An integer corresponding to the atlas of interest. 1=Desikan, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400. Set to 1 by default. This argument is ignored for hippocampal surfaces.
mode	A string indicating whether to extract the sum ('sum') or the average ('mean') of the ROI vertices values. Default is 'mean'.

Details

The function currently works with the aparc/Desikan-Killiany-70, Destrieux-148, Glasser-360, Schaefer-100, Schaefer-200, Schaefer-400 atlases. ROI to vertex mapping data were obtained from the 'ENIGMA toolbox' ; data for Destrieux came from 'Nilearn' 's nilearn.datasets.fetch_atlas_surf_destrieux

For hippocampal data, the function currently works with the "bigbrain" 10-parcels atlas integrated in 'HippUnfold.' See also doi:10.1016/j.neuroimage.2019.116328.

Value

A matrix object with ROI as column and corresponding average vertex-wise values as row

See Also

atlas_to_surf

Examples

```
CTv = runif(20484,min=0, max=100)
parcel_data = surf_to_atlas(CTv, 1)
```

---

surf_to_vol *Surface to volume*

Description

Converts surface data to volumetric data (.nii file)

Usage

```
surf_to_vol(surf_data, filename, VWR_check = TRUE)
```

Arguments

surf_data	A numeric vector or object containing the surface data, either in fsaverage5 (1 x 20484 vertices) or fsLR32k (1 x 64984 vertices) space. It can only be one row of vertices (not a cohort surface data matrix).
filename	A string object containing the desired name of the output .nii file (default is 'output.nii' in the R temporary directory (tempdir())).
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Value

A .nii volume file

Examples

```
CTv = runif(20484,min=0, max=100);
surf_to_vol(CTv, filename = paste0(tempdir(),'/volume.nii'), VWR_check=FALSE)
```

TFCE_threshold *Thresholding TFCE output*

Description

Threshold TFCE maps from the TFCE_vertex_analysis() output and identifies significant clusters at the desired threshold.

Usage

```
TFCE_threshold(TFCEoutput, p = 0.05, atlas = 1, k = 20, VWR_check = TRUE)
```

Arguments

TFCEoutput	An object containing the output from TFCE_vertex_analysis()
p	A numeric object specifying the p-value to threshold the results (Default is 0.05)
atlas	An integer corresponding to the atlas of interest. 1=Desikan, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400. Set to 1 by default. This argument is ignored for hippocampal surfaces. For applicable SubCortexMesh surfaces, 1=default base ROI, 2=anatomical atlas.
k	Cluster-forming threshold (Default is 20)
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Value

A list object containing the cluster level results, unthresholded t-stat map, thresholded t-stat map, and positive, negative and bidirectional cluster maps.

Examples

```
model1_TFCE=readRDS(file = url(paste0("https://github.com/CogBrainHealthLab",
"/VertexWiseR/blob/main/inst/demo_data/model1_TFCE.rds?raw=TRUE")))

TFCEanalysis_output=TFCE_threshold(model1_TFCE, p=0.05, atlas=1,
VWR_check=FALSE)
TFCEanalysis_output$cluster_level_results
```

`TFCE_vertex_analysis` *Vertex-wise analysis with threshold-free cluster enhancement (fixed effect)*

Description

Fits a linear model with the cortical or hippocampal surface data as the predicted outcome, and returns t-stat and threshold-free cluster enhancement (TFCE) statistical maps for the selected contrast.

Usage

```
TFCE_vertex_analysis(
  model,
  contrast,
  formula,
  formula_dataset,
  inverse = FALSE,
  surf_data,
  nperm = 100,
  tail = 2,
  nthread = 10,
  smooth_FWHM,
  VWR_check = TRUE
)
```

Arguments

`model`	An N X P data.frame object containing N rows for each subject and P columns for each predictor included in the model
`contrast`	A N x 1 numeric vector or object containing the values of the predictor of interest. Its length should equal the number of subjects in model (and can be a single column from model). The t-stat and TFCE maps will be estimated only for this predictor.
`formula`	An optional string or formula object describing the predictors to be fitted against the surface data, replacing the model, contrast, or random arguments. If this argument is used, the formula_dataset argument must also be provided. • The dependent variable (DV) is not needed, and the formula will start with ~. The DV will be the surface data value by default, but it can be swapped with contrast as IV via the "inverse" argument. • The first independent variable in the formula will always be interpreted as the contrast of interest for which to estimate cluster-thresholded t-stat maps. • Only one random regressor can be given and must be indicated as '(1|variable_name)'.
`formula_dataset`	An optional data.frame object containing the independent variables to be used with the formula (the IV names in the formula must match their column names in the dataset).

inverse	A boolean object stating whether to set the surface data as predictor of the contrast variable, instead of as dependent variable (default is FALSE). Other covariates in the model remain independent variables. This makes modelling slower.
surf_data	A N x V matrix object containing the surface data (N row for each subject, V for each vertex), in fsaverage5 (20484 vertices), fsaverage6 (81924 vertices), fslr32k (64984 vertices) or hippocampal (14524 vertices) space. See also Hipvextract(), SURFvextract(), FSLRvextract(), or SCMvextract() output formats. Alternatively, a string object containing the path to the surface object (.rds file) outputted by extraction functions may be given.
nperm	An integer specifying the number of permutations generated for the subsequent thresholding procedures (default = 100)
tail	An integer specifying whether to test a one-sided positive (1), one-sided negative (-1) or two-sided (2) hypothesis
nthread	An integer specifying the number of CPU threads to allocate
smooth_FWHM	A numeric vector specifying the desired smoothing width in mm. It should not be specified if the surf_data has been smoothed previously with smooth_surf(), because this results in surf_data being smoothed twice.
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Details

This implements the Freedman–Lane nuisance-adjustment strategy used by FSL's randomise, while adapting the TFCE method from the 'Nilearn' Python library.

A reduced linear model is fitted using an intercept and all nuisance predictors, excluding the contrast. For each permutation, residuals from this reduced model are shuffled across observations and added back to its fitted values. The full linear model is then refitted to obtain the t-statistic for the tested predictor at each vertex.

Value

A list object containing the t-stat and the TFCE statistical maps which can then be subsequently thresholded using TFCE_threshold()

See Also

RFT_vertex_analysis, TFCE_vertex_analysis_mixed, TFCE_threshold

Examples

```
demodata = readRDS(system.file('demo_data/SPRENG_behdata_site1.rds',
package = 'VertexWiseR'))[1:5,]
CTv = readRDS(file = url(paste0("https://github.com",
"/CogBrainHealthLab/VertexWiseR/blob/main/inst/demo_data/",
"SPRENG_CTv_site1.rds?raw=TRUE")))[1:5,]

TFCEpos=TFCE_vertex_analysis(model=demodata[,c("sex","age")],
contrast=demodata[,"age"], surf_data=CTv, tail=1,
```

```
nperm=5, nthread = 2, VWR_check=FALSE)

#To threshold the results, you may then run:
#results=TFCE_threshold(TFCEpos, p=0.05, atlas=1)
#results$cluster_level_results

#Formula alternative:
#formula= as.formula("~ age + sex")
#TFCEpos=TFCE_vertex_analysis(formula=formula,
#formula_dataset=demodata, surf_data=CTv, tail=1,
#nperm=5, nthread = 2, VWR_check=FALSE)
```

`TFCE_vertex_analysis_mixed` *Vertex-wise analysis with threshold-free cluster enhancement (mixed effect)*

Description

Fits a linear mixed effects model with the cortical or hippocampal surface data as the predicted outcome, and returns t-stat and threshold-free cluster enhancement (TFCE) statistical maps for the selected contrast. Permutation

Usage

```
TFCE_vertex_analysis_mixed(
  model,
  contrast,
  random,
  formula,
  formula_dataset,
  inverse = FALSE,
  surf_data,
  nperm = 100,
  tail = 2,
  nthread = 10,
  smooth_FWHM,
  perm_type = "within_between",
  VWR_check = TRUE
)
```

Arguments

`model`	An N X P data.frame object containing N rows for each subject and P columns for each predictor included in the model.This data.frame should not include the random effects variable.

`contrast` A N x 1 numeric vector or object containing the values of the predictor of interest. Its length should equal the number of subjects in model (and can be a single column from model). The t-stat and TFCE maps will be estimated only for this predictor.

`random` A N x 1 numeric vector or object containing the values of the random variable (optional). Its length should be equal to the number of subjects in model (it should NOT be inside the model data.frame).

`formula` An optional string or formula object describing the predictors to be fitted against the surface data, replacing the model, contrast, or random arguments. If this argument is used, the formula_dataset argument must also be provided.

- The dependent variable (DV) is not needed, and the formula will start with ~. The DV will be the surface data value by default, but it can be swapped with contrast as IV via the "inverse" argument.
- The first independent variable in the formula will always be interpreted as the contrast of interest for which to estimate cluster-thresholded t-stat maps.
- Only one random regressor can be given and must be indicated as '(1|variable_name)'.

`formula_dataset` An optional data.frame object containing the independent variables to be used with the formula (the IV names in the formula must match their column names in the dataset).

`inverse` A boolean object stating whether to set the surface data as predictor of the contrast variable, instead of as dependent variable (default is FALSE). Other covariates in the model remain independent variables. Formula cannot be modified (with surf_data as predictor) to trigger inverse. This makes modelling substantially slower.

`surf_data` A N x V matrix object containing the surface data (N row for each subject, V for each vertex), in fsaverage5 (20484 vertices), fsaverage6 (81924 vertices), fslr32k (64984 vertices) or hippocampal (14524 vertices) space. See also Hipvextract(), SURFvextract(), FSLRvextract(), or SCMvextract() output formats. Alternatively, a string object containing the path to the surface object (.rds file) outputted by extraction functions may be given.

`nperm` An integer specifying the number of permutations generated for the subsequent thresholding procedures (default = 100)

`tail` An integer specifying whether to test a one-sided positive (1), one-sided negative (-1) or two-sided (2) hypothesis

`nthread` An integer specifying the number of CPU threads to allocate

`smooth_FWHM` A numeric vector object specifying the desired smoothing width in mm. It should not be specified if the surf_data has been smoothed previously with smooth_surf(), because this results in surf_data being smoothed twice.

`perm_type` A string object specifying whether to permute the rows ("row"), between subjects ("between"), within subjects ("within") or between and within subjects ("within_between") for random subject effects. Default is "row".

`VWR_check` A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Details

This implements the Freedman–Lane nuisance-adjustment strategy used by FSL's `randomise`, while adapting the TFCE method from the 'Nilearn' Python library.

The permutation procedure extends the Freedman–Lane approach to the homogeneous random-intercept model fitted by this function. A reduced mixed model is fitted after removing the `contrast` while retaining all nuisance fixed effects and the original random-intercept structure.

Marginal residuals are calculated by subtracting the reduced model's fixed-effects fitted values from the outcome. Predicted random effects are not subtracted, so the residuals retain the random-effect contribution. The full mixed model is then refitted, including re-estimation of its variance components.

Value

A list object containing the t-stat and the TFCE statistical maps which can then be subsequently thresholded using TFCE_threshold()

See Also

`RFT_vertex_analysis`, `TFCE_vertex_analysis`, `TFCE_threshold`

Examples

```
demodata = readRDS(system.file('demo_data/SPRENG_behdata_site1.rds', package = 'VertexWiseR'))[1:5,]
CTv = readRDS(file = url(paste0("https://github.com",
"/CogBrainHealthLab/VertexWiseR/blob/main/inst/demo_data/",
"SPRENG_CTv_site1.rds?raw=TRUE")))[1:5,]

TFCEpos=TFCE_vertex_analysis_mixed(model=demodata[,c("sex",
"age")], contrast=demodata[,"age"], random=demodata[,"id"],
surf_data=CTv, nperm =5,tail = 1, nthread = 2, VWR_check=FALSE)

#To get significant clusters, you may then run:
#results=TFCE_threshold(TFCEpos, p=0.05, atlas=1)
#results$cluster_level_results

#Formula alternative:
#formula= as.formula("~ age + sex + (1|id)")
#TFCEpos=TFCE_vertex_analysis_mixed(formula=formula,
#formula_dataset=demodata, surf_data=CTv, tail=1,
#nperm=5, nthread = 2, VWR_check=FALSE)
```

`VWRfirstrun` *VertexWiseR system requirements installation*

Description

Helps the user verify if VertexWisrR's system requirements are present and install them (a suitable 'Python' or 'Miniconda' environment, 'BrainStat' toolbox and libraries). If they are installed already, nothing will be overwritten.

Usage

```
VWRfirstrun(requirement = "any", n_vert = 0, promptless = FALSE)
```

Arguments

`requirement`	String that specifies a requirement to enquire about: • For only Python/Conda installation: 'python/conda only' • For Python/Conda and Brainstat installation: 'conda/brainstat' • For specific 'BrainStat' libraries: 'fsaverage5', 'fsaverage6', 'fslr32k', 'yeo_parcels' • For the neurosynth database: 'neurosynth'. Default is 'any' and checks everything.
`n_vert`	Numeric vector indicating the number of vertices of a given surface data so that only the required templates are asked for. It will modify the requirement argument accordingly.
`promptless`	A boolean object specifying whether to prompt the user for action when system requirements are missing. If TRUE, VWRfirstrun() will simply inform of what is missing and will not prompt for action. Default is FALSE.

Details

VertexWiseR imports and makes use of the R package 'reticulate.' 'reticulate' is a package that allows R to borrow or translate Python functions into R. Using 'reticulate', the package calls functions from the 'BrainStat' Python module. For 'reticulate' to work properly with VertexWiseR, a Python environment needs to be installed with it — the default choice offered by VWRfirstrun is to let reticulate (version 1.41.0) create an ephemeral Python virtual environment using UV and py_require(). If for a reason this is not desirable, VWRfirstrun() also gives the choice to install a 'Miniconda' (lightweight version of Python) or Python environment in a reticulate default path or a specified path. Vertex-wise statistical analyses of cortical surface require fsaverage and parcellation templates as imported by default in 'BrainStat'. The decode_surf_data() function also requires the 'Neurosynth' database to be downloaded.

Value

No returned value in interactive session. In non-interactive sessions, a string object informing that system requirements are missing.

Examples

```
VWRfirstrun()
```

Index

∗ **datasets**
edgelist_hip, 7
fs6_to_fs5_map, 10
fslr_to_fs5_map, 12
hip_points_cells, 15
ROImap_fs5, 25
ROImap_fs6, 26
ROImap_fslr32k, 26
ROImap_hip, 27

atlas_to_surf, 3, *32*

CAT12vextract, 4

decode_surf_data, 5
DTSERIESvextract, 6

edgelist-class, 7
edgelist_hip, 7

fs5_to_fs6, 8, *9*
fs5_to_fslr, 8, *12*
fs6_to_fs5, *8*, 9
fs6_to_fs5_map, 10
fs_stats, 13
fslr_to_fs5, *9*, 12
fslr_to_fs5_map, 12
FSLRvextract, 10

getting_started, 13

hip_points_cells, 15
HIPvextract, 14

MNIsurface-class, 15

plot_overlay_surf, 16
plot_surf, 18
plot_surf3d, 20

RFT_vertex_analysis, 22, *35*, *38*
ROImap-class, 25
ROImap_fs5, 25
ROImap_fs6, 26
ROImap_fslr32k, 26
ROImap_hip, 27

SCMvextract, 27
smooth_surf, 29
surf_to_atlas, *3*, 31
surf_to_vol, 32
SURFvextract, 30

TFCE_threshold, 33, *35*, *38*
TFCE_vertex_analysis, *24*, 34, *38*
TFCE_vertex_analysis_mixed, *24*, *35*, 36

VWRfirstrun, 38